

FUJI FACTS

Newsletter of
The Atari Computer Enthusiasts of Columbus

Vol 5, No 9

September, 1987

Price: \$1.00



THE EDITOR'S COLUMN

BY: WARREN LIEUALLEN

Another new format this month! If all works as planned, and if you are satisfied, I think Fuji Facts will finally settle down and stay like this for a while! The larger sized edition is still available as well.

The topic for this month seems to be "Bits and Pieces", as I go through and collect a little of this and a little of that. Another Charles Brown review, left over from last month's issue is featured. A very informative article on The Duplicator by Jeff Moore echoes many of the opinions I have read about this hardware add-on. Chris Crawford continues his assembly language tutorial, after a respite of several months. Charles Brown comes through again with an information packed article on all the graphics modes available on the eight-bit Atari computers. Finally, there are also several interesting tidbits from our Exchange Newsletters; little things that I've been meaning to run for some time now.

Officer nominations will be upon us this month. Please, consider nominating someone deserving, or perhaps even better, consider running yourself. The club needs some new blood to keep our enthusiasm level high. Being an ACEC officer really doesn't take much of your time (with the exceptions of Disk Librarian and Newsletter Editor), and is both informative and rewarding. Where else can you have so much fun so inexpensively?!

(This paragraph is excerpted from the July/August 1987 issue of the MIA Atari Limited Edition, and expresses an editorial policy which is found throughout the country [most importantly, in Columbus, Ohio!].) It doesn't make a whole lot of sense to waste the club's money reprinting articles from others' newsletters that you can easily check out from our Publication Library. Therefore, reprints will be kept to an absolute minimum (as will Antic On-Line material). In addition, there will be no more pleading and wheedling for article submissions. I have said it too often, with too little response. So, from now on, what I get from you will be what you see, as much or as little of it as there is. An eight page newsletter makes my job a cake-walk. I look forward to you making my job a real pain in the interface!

Table of Contents:

<i>The Editor's Column</i>	1
<i>The Print Shop Interface</i>	3
<i>Assembly Language Tutorial</i>	5
<i>The Duplicator 1080</i>	7
<i>Atari BASIC Graphics</i>	9
<i>A.C.E.C. Meeting Minutes</i>	11

ATARI COMPUTER ENTHUSIASTS

OF COLUMBUS
ESTABLISHED 1982

This newsletter is written and published monthly by the Atari Computer Enthusiasts of Columbus (ACEC). ACEC is an independent, non-profit organization interested in exchanging information about any and all Atari Home Computer Systems.

Our main meetings are held on the second Monday of each month at 7:15 p.m., at DeSales High School (on Karl Road, just south of Morse Rd.), and are open to the public. Other Special Interest meetings are held as announced at the main meeting.

Dues are \$12.00 per year, and entitle members to all club benefits (Newsletter, Disk of the Month, Publications Library, SIG meetings, group discounts at selected area merchants, etc.).

Fuji Facts welcomes contributions of articles, reviews, editorials and any other material relating to the Atari computers, or compatible hardware devices and software packages.

ACEC is not directly affiliated with the Atari Corporation; "ATARI" and the "Fuji" symbol are registered trademarks of the Atari Corporation. All other trademarks, copyrights and service marks are registered with their respective owners.

The statements expressed in this newsletter are solely the opinions of the authors, and do not necessarily reflect those of ACEC, its officers or its members. Material contained in this newsletter may be reprinted provided credit is given to both Fuji Facts and the author(s).

The cover and headers of this month's newsletter were printed with a Star SG-10 dot matrix printer, using TypeSetter 130 and The Print Shop. The newsletter itself was printed with a Star PowerType daisywheel printer in condensed pitch, using PaperClip 2.0 on a 256K RAM modified Atari 800 XL.

Our permanent mailing address is:
P.O. Box 849
Worthington, OH 43085

PRESIDENT:

Charles Lusco
4624 Channing Terrace, #C
Columbus, OH 43232
863-4016

VICE-PRESIDENT:

Dave Beck
1194 Country Club Road
Columbus, OH 43227
863-8600

PUBLICATIONS LIBRARIAN:

Mark Schmidbauer
1980 Belcher Drive, Apt. C3
Columbus, OH 43224
262-5804

MEMBERSHIP CHAIRMAN:

Position currently
vacant

NEWSLETTER EDITOR:

Warren Lieuallen
1652 Hess Boulevard
Columbus, OH 43212
488-3977

DISK LIBRARIAN:

Jim Murphy
291 Millside Drive
Gahanna, OH 43230
476-3751

TREASURER:

Dave Feeney
2665 Blue Rock Boulevard
Grove City, OH 43123
871-0524

SECRETARY:

Don Bowlin
230 Orchard Lane
Columbus, OH 43214



THE P.S. INTERFACE

BY: CHARLES BROWN



This is the last in my series on desktop publishing and printer art work. I hope in these articles that I have given you a glimpse into what these programs are like. If you have a printer that can handle graphics, you can really enjoy all the nice features of your Atari computer. Being able to put your artistic creations on paper can be of great enjoyment to everyone.

This program is called PS Interface. It is part of the XLEnt family of software. This program is a collection of conversions utilities. They are used to covert files from one format to another. It is useful to convert from Print Shop data to TypeSetter format, and vice versa.

When you first load in the program you will be greeted with a four choice menu. The four choices are: font creator; graphic conversion; font conversion; and exit to DOS for standard disk operation.

Put your artistic creations on paper...

The first choice is a font creator. This is used to change one of the original eight fonts on the back of the Print shop disk. It is a very complicated procedure. I am not to thrilled with it at all, so I am going to try and explain why. You first must make a copy of the B side of the Print Shop disk. This is so you don't mess up the original. The manual tells you to use a regular drawing program and draw the characters in your font. I am sure that the people from XLEnt have never seen my attempts at art! I tried to use this feature, so I used The Atari Touch Tablet and drew my own little font. Unlike most font editors, you can have more than one character on the screen at a time. In my example, I had the letters A-X in one file and Y-Z in the second file. It had to be saved in an uncompressed (MicroPainter format. This means that the file must be 62 sectors long in order to work with PS interface. So I now have two files of my custom characters to use.

I can now go back to PS Interface and load in the font creator program again. I am now presented with another menu to work with. The first choice is a disk directory for dumb people like me who can't remember the filenames. The next feature is to load font pictures. This is where I load my files from the touch tablet. The third option is to load a TypeSetter icon. It can have a picture in it which can be used in your new font. If I understand it correctly, a picture (icon) can even replace a letter, but I am not sure.

The next feature is the hardest. This is where you actually make the changes in the font. It is a very long and drawn out process. When you pick this feature you first pick out the character that you want changed. Let's say that I want to make a new letter A, so I type it in. Then I am given the graphic page. This shows the screen that I made with the touch tablet that contains my custom letters A-X. I have a small cursor in the middle of the screen that I move around with a joystick. I have to move this little cursor over to the upper left hand corner of the character that I want my A to look like. When I push the joystick button I have a box that I have to surround my character with. When I push the button the second time the character in the box is my new letter A. If I want to change all the characters I am going to be here a long time! I would also have to load in my second file of characters that I made with the touch tablet.

The next to last feature is to save your new font to the Print Shop disk. In this mode you want to use your copy of the back side of it. You put the copy disk in your drive and you will be asked which font do you want to replace. This is one of the original eight supplied by Print Shop. Your new font will replace one of the original fonts. The manual tells you that you should try and keep your fonts around the same size as the ones you're replacing. The very last feature is to erase the font in memory. This is so you can start over with a clean slate.

The next selection from the main menu is the graphic converter utilities. This usually is the main reason that people get this program. With the graphic converters, you can do a lot of switching around between the two graphic programs (Print Shop and TypeSetter). When this program is loaded in you will be given a menu

of choices. There are two columns in this menu. The choices in the left column are for operations that involve the Print Shop as the destination disk. So, if you are switching something over to Print Shop format then you would use the choices on the left side of the menu.

On the other hand, the choices in the right column are for operations that involve a standard DOS disk as the destination. If you are switching something from Print Shop format then you would use the choices on the right side of the menu.

Both sides of the menu give you a directory option, which will show you your files on the disk. You are also given format options in both modes. You can either format a disk with either a standard DOS or Print Shop style. The Print Shop side also gives you a chance to either delete or rename files. The rest of the options are for converting files.

The first option on the left side (going to Print Shop) is to convert from any uncompressed file (62 sectors) to Print Shop. This picture is to be used in the Screen Magic part of Print Shop.

The next feature is Graphic to Print Shop. This allows you to change a picture (or part of one) to a Print Shop graphic. If you choose to change just a part of a picture, you will first load in the picture, and then see your picture with a frame on it. You move the frame with a joystick to enclose the part of the picture that you want to convert. If you choose to convert the whole picture, it will be compressed to Print Shop format. You will see twelve different versions of your converted picture. You choose which graphic you want to use with the keyboard.

The last option on the Print Shop side is TypeSetter to Print Shop. It is an almost exact copy of the feature explained above. The only difference is that you will be given a choice of only three versions of the picture instead of twelve.

The first conversion choice on the right side of the menu is screen to DOS II. This feature lets you take an uncompressed Print Shop picture and change it to a standard DOS style. Now you can use it in an art program or a screen dump program.

The last feature is graphic to DOS. This part will take a picture on a Print Shop disk and turn it into a TypeSetter icon. These conversions can give you several choices to pick from.

The last major option from the main menu is a font converter. This utility will change a Print Shop font over to a TypeSetter style. With this conversion, each letter will be saved as a separate file with predefined names. If you want to change the whole font, it will take up a complete side of your disk. You are given four choices for sizes of the characters. If you wanted to save off all eight of the different original Print Shops in the four different sizes, you'd better have 32 blank disks handy!

Once you have your fonts converted over to TypeSetter, you have to load in the Sketch Pad part of TypeSetter. You then have to load in each letter one at a time, and stamp it on to the main screen. It sounds like a very long process, because it is!

I am somewhat happy with this program. I feel that the graphic conversion parts are very useful. I can see where they will come in very handy. The part that changes fonts to Print Shop would be handy if you want to change pictures into letters, but I'm not sure just why you'd want to do this. The part that changes Print Shop fonts over to TypeSetter format I don't like at all, especially if you have to load in each letter one at a time. I could almost draw my own letters faster than that.

★ **WANTED:** ★
DEAD OR ALIVE

**REGULAR COLUMNISTS
FOR FUJI FACTS.**

WRITE ABOUT:

**BASIC
ACTION!**

**ASSEMBLY LANGUAGE
C**

ANYTHING ELSE!

**(DEAD COLUMNISTS
NEED NOT APPLY)**



ASSEMBLY LANGUAGE TUTORIAL



BY: CHRIS CRAWFORD

ANTIC Publishing Inc.
copyright 1985. Reprinted by
permission from Antic OnLine.

LESSON SIX: SUBROUTINES & THE STACK

We now take up the first topic in this series that is not absolutely essential to writing programs: subroutines. The loops and indexed addressing discussed in the previous lecture are truly essential: it is hardly possible to write a useful program that has no loops. Subroutines are a matter of convenience, not necessity.

It is quite possible to write an entirely adequate program without using a single subroutine. However, you will find that the convenience of subroutines with large programs is so great that you would never want to write such a program without them.

The primary purpose of a subroutine is to perform some function that is frequently needed at many points in the program. Instead of having to repetitively insert the same code over and over again, we simply write it once, place it in a subroutine, and call that subroutine many times from the main program. The use of subroutines dramatically reduces the size of a program.

Subroutines are implemented on the 6502 in a fashion very similar to that used by BASIC. You may recall the two BASIC commands for subroutines: "GOSUB lineno" and "RETURN". The two corresponding 6502 commands are "JSR label" and "RTS". The label in "JSR label" is the label of the beginning of the subroutines.

Thus, writing and using subroutines in 6502 is trivially simple. First, you write the subroutine. You give it a name (say, "MYSUBR") and stick that label in front of the first instruction. You put an RTS command after the last normal command of the subroutine. To call the subroutine, you just put JSR MYSUBR. That's all it takes!

However, in order to understand how it works is not so easy. Here's the problem we must solve when the 6502 jumps to a subroutine. The JSR instruction tells it the destination address to which the 6502 must jump. But when the 6502 hits the RTS instruction, how does it know the address to which it must return?

The RTS doesn't say, "Return to THIS address"; it says only "Return". Moreover, how could the 6502 know where to return? If the subroutine can be called from, say, five different points in the program, how would the 6502 know which of those points to which it must return?

What if we gave the 6502 a special register for remembering return addresses? That is, whenever the 6502 encounters a JSR instruction, it stores the current address into its return address register. Then when it encounters an RTS instruction, it simply takes the address out of the return address register.

Subroutines are a matter of convenience, and are not a necessity.

There is only one problem with this: what if we use nested subroutines (one subroutine calls another)? The second subroutine call will erase the return address for the first subroutine call. Trouble!

The solution to all this is called a stack. A stack is a chunk of RAM allocated for certain special operations such as subroutines. The 6502 stack is stored on page one — that is, addresses \$0100 to \$01FF. The stack operates like 128 return address registers arranged in sequence (remember: two bytes per address). The 6502 keeps a stack pointer register to keep track of which byte in the stack is currently being used.

I will now trace through the operation of the stack in a subroutine. We start with the stack pointer set equal to \$FF. That means that the stack is empty; the stack pointer is at the very top of the stack. The 6502 encounters a JSR instruction. It takes the current value of the program counter and breaks it into two bytes. It pushes the first byte onto the stack. This means that it stores the first byte at \$01FF, then decrements the stack pointer. Now the stack pointer is \$FE.

Then the 6502 then pushes the second byte of the return address onto the stack, storing that byte at \$01FE and decrementing the stack pointer to \$FD. Then the 6502 jumps to the subroutine. When it encounters the RTS instruction, it pulls the two address bytes off of the stack (increments stack pointers and loads byte at address \$0100, SP). Those two bytes go directly into the program counter, returning the 6502 to the original entry point.

The advantage of this approach is that it allows very deep nesting of subroutines. If one subroutine calls another, the 6502 simply stores more values onto the stack. The addresses won't be confused because you always exit subroutines in exactly the reverse of the order that you entered them.

You can use the stack yourself, if you wish. You have six instructions that allow you to play with the stack: PHA, PLA, PHP, PLP, TSX, and TXS.

The PHA instruction pushes the value of the accumulator onto the stack and decrements the stack pointer. The PLA instruction increments the stack pointer and pulls the current stack value into the accumulator. These two instructions allow you to store and retrieve values onto the stack. They must be exactly balanced, though, or you will generate that most feared of bugs, the stack crash.

Consider: you are in a subroutine. You push a value onto the stack, but forget to pull it off. When the 6502 attempts to return to its original location, it pulls two address bytes off the stack — but they're the wrong two bytes. One of them is the value you pushed but didn't pull. Result: the 6502 return to the wrong address. Your program goes haywire and the computer crashes.

This is called a stack crash. This type of crash tends to be particularly difficult to recover from. Prevention is the best medicine. The rule for preventing stack crashes is simple and absolute: each and every push onto the stack must be balanced by one pull from the stack. Violate this rule and you will certainly experience a stack crash.

The next pair of stack manipulation instructions are PHP and PLP. These push and pull the process status register from the stack. They are useful for two purposes. First, you may wish to save the values of the various flags before performing some operation, then restore them so that you can branch on a previously created condition. Second, it is sometimes handy to PHP, then PLA to get the processor status register into the accumulator where you can more directly manipulate it. Again, each push must be balanced by one pull.

The third stack manipulation pair of commands do not modify the stack. They are TSX and TXS. These transfer the stack pointer to and from the x-register. Once in the x-register, you can change the value of the stack value and then TXS to jump over sections of the stack. This can be a very handy way to pass parameters to subroutines, but it is also very tricky. If you make a mistake, you will generate a stack crash. So be careful with this one. I have always avoided these commands — the plague. They are very dangerous and never esse

reprinted from The End User
by Arthur Leyenberger
Analog Computing, April 1987

New QMDS 6502 Chip

It has just been announced that a new low-power version of the venerable 6502 microprocessor will be included in all new Atari XE computers. The 65002 chip is completely bug compatible with the existing 6502, and offers a few additional op-codes for use by serious programmers. These op-codes should finally allow serious business software to be written for the eight-bit Atari, helping to change the unfounded "game" image the computer currently enjoys. Listed below are just a few of these new op-codes:

BNR — Branch for No Reason
CMD — Create Meaningless Data
DRA — Decrement Random Access
EDR — Emit Deadly Radiation
ICR — Incur Costly Repair
JIL — Jump to Inaccessible Location
LLI — Lose Last Instruction
PRS — Push Results off Stack
RIS — Remain In Subroutine
SHB — Scramble High-order Bit
TEC — Take Extra time for Calculation



THE DUPLICATOR 1050

BY: JEFF MOORE



A couple of years ago, I began looking for a program to create backups for my heavily used programs — those which I have purchased and would hate to lose through equipment breakdown or any other unexpected disk destroying catastrophe.

After searching for the right program, I found the DUPLICATOR and HAPPY disk drive hardware add-ons. The DUPLICATOR sounded so much better and was cheaper than the HAPPY. I sent for the DUPLICATOR and in short order it arrived at my door. The installation instructions sounded simple enough, so I dove right into it. Two hours later, it was installed and the moment of truth had arrived. A simple flick of the switch revealed however, that it didn't work. Checking the wiring and experimenting, I found that I had a connection on backwards. This was easy to do because they don't fit on just one way. The drive motor now ran and shut off normally, but nothing would load.

I called their toll phone number in New York for technical assistance and was told, "Sounds like your main chip is bad. We'll send you a new one because we've had problems with brown outs while we were making some of them." I disassembled the drive again and removed the DUPLICATOR (replacing the 2 factory installed chips), and reassembled it again. My Atari disk drive was back again!

When the main EPROM arrived and everything was torn apart and installed again, it was discovered that the 'main' chip in my original DUPLICATOR wasn't even there to begin with, only its socket. Once replaced and put back together again, everything seemed to work. I could even copy some disks (but not all the ones promised in their advertisement). My drive could even read and write faster and even warp speed with the proper software! SpartaDOS was included free with the package, but it included no instructions! Have you ever sat there and waited for your drive motor to stop after a read or write? Well, now I can set the time it takes to turn the motor off.

I ran across a couple of new programs that wouldn't even boot in slow (normal Atari) mode, but would run on unmodified drives. My DUPLICATOR 1050 drive would frequently get "confused", either while reading or writing, and would get a lot of disk errors and device

timeouts. It could still format single density, but forget about enhanced and sometimes double densities. I called DTI again, with no satisfaction.

A few months later a new EPROM (Rev 3) and software (Rev 2) were sent to me to solve some of the "bugs" concerning formatting and double density. They asked that the old chip and software disk be sent back, as they wouldn't be of any further use to me (Tear the drive apart again!). The next revision was to be FREE and was to bring the DUPLICATOR up to its full potential.

Next, a letter arrived and announced the new Revision #3. The paper instructed me to send back my old software, including the main chip and they would send out the new revision free, or just send \$10 instead. I wrote a letter to them explaining that I didn't want to lose the use of my drive while I waited and promised I'd send the old ones back after receiving the replacements. No response! I sent the ten dollars.

I tore the drive apart again and still it wouldn't copy all of the promised software, but it would format now. I still got some device done errors and time out errors and got a bonus. It would now occasionally SCRAMBLE the disk directory. After ruining a couple of disks I wrote them again and went out and bought a used Indus.

Another letter arrived wanting \$16.50 for another enhancement (Rev 4) that would copy more programs, but still not all of the ones promised in their original advertisement. I ordered the Happy Cheerup upgrade for the DUPLICATOR. \$50 was better than to keep sinking money into a useless DUPLICATOR. Happy promised that theirs would duplicate their Happy drive for use with any of their software to copy all promised software.

DUPLICATOR is now up to Revision 5 (\$17.50). All of these in about two years and I'll bet that the DUPLICATOR still doesn't work right!

The earlier DUPLICATOR was easy to use. Boot the disk and follow the directions. The program would even ask how many times it should try to copy a given track before going to the next (default was 8). It even printed the type of protection that it found on each track. Many times, you wouldn't know that it couldn't

copy a given program until you ran it afterwards. They did include a partial list of software it could copy and using which revision (current or future). The DUPLICATOR would even ask after copying if you wanted to boot the copy and in regular or forced slow modes.

Some copy protection involves 19 or 20 sectors per track so another device had to be installed (included in original package) to slow the motor down to slower than normal. This was activated by the copy program software to copy this form of protection.

Revision 3 even included a program called "format fix" for solving problems formatting double density. Some games had to go through two copy programs to work, the normal copy program and a custom format program (for 10 programs). If you think your copy wouldn't run because of track skew alignment, there is another program for that.

I haven't had the Happy Cheerup for very long, but I haven't had any major problems with it. The first copy of the software wasn't complete and upon calling them it was promptly replaced. My disk drive hasn't been confused, no device time out, no formatting problems, no scrambled directories, etc. There is a list of companies, their games and utilities, and complete instructions on how to copy and run them (some have to be run on a Happy drive whereas the DUPLICATOR did not). Four out of five programs that the DUPLICATOR couldn't copy were copied by the Happy. This was the first time that I tried the HAPPY and I've tried copying only once. It works in all densities and has warp speed capacity. It is supported in CompuServe, too.

In summary, DO NOT WASTE YOUR MONEY ON A DUPLICATOR!!

T.O.S. ERROR LISTING

BY: ALLEN BARGEN

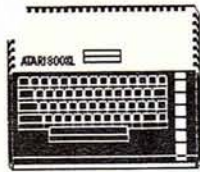
REPRINTED FROM MILATARI LIMITED EDITION, JULY/AUG. 1987

TOS'ing and Turning All Night

A growing number of ST owners have been asking about the meaning of those annoying TOS error messages you get when everything does not go according to plan. Here is an annotated listing of the most commonly encountered ones that should help:

- 0 = A-OK (no problem)
- 1 = General Error
- 2 = Drive Not Ready (device not ready, not attached or has been busy too long)
- 3 = Unknown Command (device did not understand you)
- 4 = CRC Error (soft read error)
- 5 = Bad Request (device could not handle the command)
- 6 = Seek Error (drive could not perform a seek)
- 7 = Unknown Media (attempt to read an unformatted sector or foreign media)

- 8 = Sector Not Found (requested sector not found)
- 9 = No Paper (printer is out of paper)
- 10 = Write Fault (a write operation failed)
- 11 = Read Fault (a read operation failed)
- 12 = General Mishap (reserved for future use)
- 13 = Write Protect (the disk is write protected)
- 14 = Media Change (media has been changed since the last write)
- 15 = Unknown Device (operation specified a device that the BIOS couldn't recognize)
- 16 = Bad Sectors (format operation has detected bad sectors)
- 17 = Insert Disk (request for you to insert disk)



ATARI BASIC GRAPHICS

BY: CHARLES BROWN



I have heard that a number of ACEC members are interested in using graphics on their computers. I have always wanted to be an artist, and I have said before that I probably have almost all the artistic programs for the 8 bit machine. I still can't draw anything decent, although I enjoy trying. I always enjoyed writing graphic programs in BASIC. I even donated some to our Disk of the Month. Since that time I have gotten into machine language programming. I am now starting to try my luck at graphics in machine language. I will enjoy getting the speed that machine language programming has over BASIC. In case there is anyone that is interested, I will try and give a little insight into the graphics capability of your Atari.

To start off, there are different graphic modes that your computer uses. Think of these modes as being different screens. How many different modes you have available depends on what computer you have. If you have an older model 400 or 800, then you probably have a GTIA chip in your computer. This would mean that you only have nine different graphic modes available from BASIC. The only way to get any more is to use machine language. If you have a newer model computer (XL or XE) you probably have the new GTIA chip in your computer. This will give you three extra modes in basic for a total of twelve. As mentioned above, no matter what chip you have, there are other graphic modes that your computer uses. You can't directly get them from BASIC. In order to use them you have to use machine language.

When we talk about the different graphic modes they are usually described in two different ways. The first is the size of the screen. The second way is the number of colors that can be used in that mode. When we talk about the size of the screen, you probably will hear two numbers mentioned. The first number is normally the number of columns going across the page from left to right. The second number will be the number of rows from top to bottom. Another way of putting this is the number of characters in a text mode or the number of points or pixels in a graphic mode. If a screen is 20 by 10, then you could type in twenty characters across the page in one line, and ten characters in a row from top to bottom. This is usually how the size of mode is described. When you hear about the different colors used in the different modes, you should remember that one of the colors is the background color. Of course, you can change it if you

choose. Another thing that you should know about the different modes are the text windows. These are little windows that you can have at the bottom of your mode 1-8 screens. These windows will give you 4 lines of a normal mode 0 text screen. This gives you the chance to have a picture on the screen with four lines of text at the bottom. You can choose to have a text window or not in all of these modes.

The first three modes of your computer are called text modes. They are used for displaying text on the screen. The first one is the most familiar. It is called graphic mode 0. It is the default graphic mode for your computer. When you turn on your computer with BASIC installed and see the familiar ready prompt, you are looking at a graphic mode 0 screen. This screen is 40 by 24 and uses three colors if you count the background. Most programs or applications use this mode. The next mode is graphic mode 1. It is used for printing larger and bolder characters. Its size is 20 by 20 with a text window or 20 by 24 without. This mode will give you four different colors of text and a separate screen color if you wish. The next mode is mode 2. This will give you even bigger letters. The size of this screen is 20 by 10 down with a text window or 20 by 12 without. It has the same color capabilities as graphic mode 1.

All the rest of the modes are graphic modes. They are for plotting and drawing. These are modes 3-8 if you have the old computer with a GTIA chip, or modes 3-11 if you have the GTIA chip. The first graphic mode is mode 3. Its size is 40 by 19 with a text window, or 40 by 23 without. This mode has four colors that you can use. The next two modes are the same size. They are 80 by 40 with a text window, or they can be 80 by 48 without. The difference between these modes is the number of colors that you can use. In mode 4 you only have two different colors. While in mode 5 you can use four different colors. The same is true for the next two graphic modes. These are modes 6 and 7. The size of both of these screens is 160 by 80 with a text window, 160 by 96 without. Again the only difference is the number of colors used. In mode 6 you only can use two colors, while in mode 7 you can use four.

Graphic mode 8 is a special mode. It has the highest resolution available. That means it has the most points that you can draw, so your drawings can have more detail. The size of it is 320 by 160 with a text window, or 320

by 192 without. You have only three colors to use in this mode, and one of those is the background, so your choices are limited. As you can see, the higher the mode number in modes 3-8, the greater the resolution. I normally use mode 7. It offers the most detail with the most colors.

I hope that this article will give you a little glimpse into the graphic capability of your computer. I tried to show you what the different modes were and what they can be used for. I hope to get some feedback from you. If I find there are people interested, then I will continue on with this subject. Again I don't claim to be an expert but if I can I will try and help you any way possible.

AT THE NEXT MEETING: PD NIGHT

COME AND SEE DEMOS
OF SOME OF THE BEST
FREE

SOFTWARE FOR ALL OF
THE ATARI COMPUTERS



GUEST EDITORIAL



BY: LEO SELL

REPRINTED FROM MICHIGAN ATARI MAGAZINE, AUG. 1987

There is a definite pall over the attitudes of Atari computer users, eight-bit users in particular. I think this attitude is part of the reason for the decline in user group membership all over the country. People get discouraged, think they have a non-supported or obsolete product, and drop out or move on. WRONG answer. While the technology is obsolete, what about the utility of the system? Does it still do what you want your computer to do? Often the answer is yes (It is for me - Ed.). Find a better reason than obsolescence to drop it or upgrade.

How about non-support? The last thing you should do is drop out or get less involved when the manufacturer of

your system and the software companies begin to drop support. On the contrary, the support responsibility is now entirely yours, and the best way to meet the responsibility is to be involved in a users' group. No one else can or will help. A users' group such as ACEC has always been the best support you can find. Now it may just be the only help you can find.

Resolve to get more involved with your Atari by redoubling your efforts and involvement in ACEC. I guarantee you'll be more satisfied than ever before.

A.C.E.C. MEETING MINUTES

AUGUST 10, 1987

The August meeting of the Atari Computer Enthusiasts of Columbus got underway at 7:15 p.m. with our usual series of short business announcements. Prominent among these were:

- 1) The newsletter is now being mailed to all PAID members about one week before the meetings. A number of people were surprised to learn that, despite the warnings printed on the mailing labels, their membership had expired. Now is the time to renew!
- 2) Although there were several minor cosmetic changes in Fuji Facts this month, I'm not done yet! Watch for next month's issue for our new, expanded and enlarged format (if I can really figure out how to do it!).
- 3) The vacancy in the position of Membership Chairman is going to be left vacant, since we are only two months away from Elections, and no one could be pressured into taking post anyway. Way to support your group, guys!

The question and answer session came next, dealing with Prometheus modems and 850 Express!, the new Atari disk drives, and more.

The main demo, following this month's theme of DeskTop Publishing, finished up. Dave Beck and Warren Lieuallen went head-to-head, both producing on their respective machines, duplicates of the cover of Fuji Facts. Dave used Easy-Draw on a 520 ST, and Warren used TypeSetter on a 130 XE. All in all, a very informative, and occasionally entertaining demo, if I do say so myself!

Don Bowlin also reminded us of the meeting dates for our monthly SIG meetings. For the rest of the year, the meetings will be held at the Whetstone Library, on: September 17, October 29, November 19 and December 17. All SIG meetings start at 7:30.

The meeting adjourned at 9:45.

Warren Lieuallen



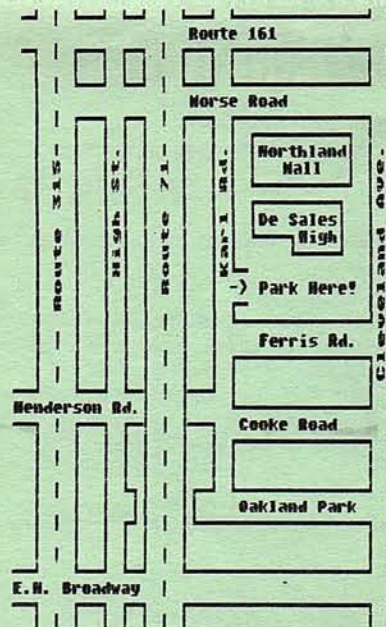
A.C.E.C. ERROR LOG



Due to some scheduling problems, there is no Disk of the Month documentation in Fuji Facts this month. However, there are a few small errors to report on last month's DOM.

- 1) The version of MicroCheck on side two is specifically for the 130 XE computers (and compatibles!). This should have been made more clear. A previous version for the 800 and 800 XL can be found on DOM # 31A2.
- 2) Both the DOS.SYS and the DUP.SYS files should be deleted, and replaced with Atari DOS 2.5 (or any other 130 XE RAMDisk-compatible DOS). Also, the RAMDISK.COM file needs to be copied to the disk.
- 3) Within the UTILITY program, the references to filenames on D8: should be changed to D1:, or you will not be able to use the program.

[not to scale]



An official Users' Group, the Atari Computer Enthusiasts of Columbus meets on the SECOND MONDAY of each month. The meetings are held at 7:15 p.m., at De Sales High School on Karl Road. Meetings are open to the public, and consist of demonstrations and short tutorials of products for the Atari Home Computer Systems. Dues for ACEC are \$12.00 per year, and include a subscription to Fuji Facts, and more!

WGL '87

Fuji Facts Newsletter
Warren Lieuallen, Editor
1652 Hess Boulevard
Columbus, OH 43212



TO:

SLCC Journal
P.O. Box 1506
San Leandro, CA 94577-0374
Last Received: 08/87

MEETING: Sep. 14th, 7:15 pm